**ALGORITHM/ PSEUDOCODE (15 points)**

1. Display welcome screen
2. Display menu options to user

Example:

1. Append a single personality test
2. Evaluate the unprocessed personality tests
3. Count and display how many people tested fell into each personality range
4. Quit
5. Ask the user to make a choice based on the menu displayed
   1. Validate the user’s menu choice
6. While the user’s choice is valid
   1. If the user’s menu choice is A or a
      1. go to Step 5
   2. if the user’s menu choice is B (or b)
      1. go to step 6
   3. if the user’s menu choice is C (or c)
      1. go to step 7
   4. if the user’s menu choice is D (or d)
      1. go to step 8
   5. else the user’s menu choice is invalid
7. Option A
   1. create the *UnprocessedTests.txt* file
   2. open the *UnprocessedTests.txt* file to be appended to
   3. Ask the user to enter their first name
      1. validate the user’s first name
      2. while the user’s entry is invalid, ask them to try again
      3. if the user’s entry is valid, break from the validation loop and continue
   4. Ask the user to enter their last name
      1. validate their last name
      2. while the user’s entry is invalid, ask them to try again
      3. if the user’s entry is valid, break from the validation loop
   5. Format the user’s name
      1. Title case the user’s first name
      2. Title case the user’s last name
      3. Concatenate the user’s firs and last name’ together and store as UserName
   6. Ask the user to enter the data for each of the 10 questions in the test (Using a for loop to ask the user for the proper question number)
8. for each of the 10 answers entered by the user validate the entry using the following parameters:
   1. the user’s entry should only contain a single character
   2. the user’s entry should be a letter from A – D
   3. the user’s entry should be case insensitive
9. while the user’s data is invalid
   1. ask them to try again
10. if the user’s entry is valid, continue
11. convert the letter to uppercase
12. Append the user’s entry to the *UnprocessedTests.txt* file
    1. Repeat step 5.e 9 more times (for a total of 10 iterations)
    2. Ask the user if they would like to return to the menu
13. Option B
    1. open the *UnprocessedTest.txt* for reading
    2. read the first line from the file and store it as the name variable
    3. while the name variable is not empty
       1. read the next line and store it as
       2. use the following key to calculate the user’s score type based on their answer for each question (using a count-controlled loop)

1. A.=3 B.=1 C.=6 D.=2  
2. A.=2 B.=1 C.=3 D.=4  
3. A.=2 B.=1 C.=3 D.=4  
4. A.=3 B.=1 C.=2 D.=6  
5. A.=1 B.=2 C.=5 D.=3  
6. A.=3 B.=1 C.=4 D.=2  
7. A.=2 B.=1 C.=3 D.=4  
8. A.=4 B.=3 C.=2 D.=5  
9. A.=6 B.=4 C.=8 D.=2  
10. A.=7 B.=5 C.=1 D.=3

* + 1. based on the score calculated in step 6.c-ii, use the following table, calculate the user’s personality type

|  |  |
| --- | --- |
| **Score** | **Type** |
| 12-20 | Personality type 1 |
| 21-30 | Personality type 2 |
| 31-42 | Personality type 3 |
| 43-53 | Personality type 4 |

* + 1. open the *ResultsTests.txt* file for appending (if it does not exist, create a new, empty one (use the exists method))
    2. append the user’s name to the *ResultsTests.txt* file
    3. append the user’s personality type to the *ResultsTests.txt* file
    4. read the next line and store it as name
    5. if the name variable is not empty repeat step 6c
    6. if the name variable is empty break from the score calculation loop
    7. clear the contents of the *UnprocessedTests.txt* file (open it for writing and close it)
    8. close the *ResultsTests.txt* file
  1. return to the menu

1. Option C
   1. open the *ResultsTests.txt* for reading
   2. read the first line in the *ResultsTests.txt* file and store it as name
   3. while name is not empty
      1. read the next line and store it as pType
      2. if pType is Personality Type 1
         1. accumulate 1 totalPersType1
      3. if pType is Personality Type 2
         1. accumulate 1 totalPersType2
      4. if pType is Personality Type 3
         1. accumulate 1 totalPersType3
      5. if pType is Personality Type 4
         1. accumulate 1 totalPersType4
      6. read the next line in the *ResultsTests.txt* file and store it as name
      7. if the name variable holds a value
         1. repeat step 7.c
      8. if the name variable is empty
         1. break from the calculation loop
   4. display the total number of each personality type

Example Output:

Personality Type 1: 0

Personality Type 2: 1

Personality Type 3: 1

Personality Type 4: 0

* 1. return the user to the menu

1. Option D
   1. thank the user for using the program
   2. ask the user if they would like to clear all the contents of the *ResultsTests.txt* file
   3. validate their answer
      1. While the user’s entry is not valid, keep repeating
      2. the entry must only be a single letter, either Y or N (case ignored)
      3. if the user’s entry is valid
         1. break from the validation loop
   4. if the user’s entry is Y (or y)
      1. clear the data in the *ResultstTests.txt* file
   5. if the user’s entry is N (or n)
      1. end program
   6. end program

**VARIABLES/IPO (2.5 points)**

**Strings**

welcome

menu

answer

choice

input

**Integers**

Score

PType

totalPersType1

totalPersType2

totalPersType3

totalPersType4

question

**Boolean**

valid

return

**FORMULAS (2.5 points)**

**TEST DATA – 5 complete data sets (10 points)**

**FLOWCHART (5 points)**

**CODE (submitted in separate compressed files) (65 points)**

*Up to a 10-point deduction for uncommented code*

*Up to a 65-point deduction if applications does not execute*